ROLL NO.- LCO1739, LCO17367 (MAYANK BIMBRA, HARSHIT)

Real Time Emotion/Facial Expressions Detection/Recognition

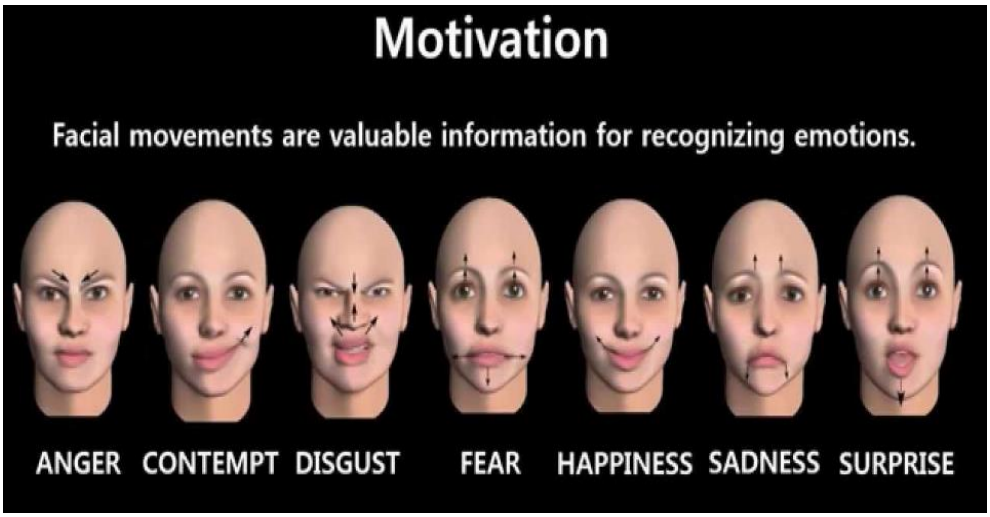
ABSTRACT

The Human facial expressions are important for visually expressing a lot more information. Facial expression recognition is essential in the field of human-machine interaction. Automated facial recognition systems have many applications, including understanding of human behavior, diagnosing mental disorders, and synthetic human expression. Identifying facial expressions through computers with high detection rates is still a challenging task.

Two of the most popular methods used in the literature for automated FER systems are geometry and appearance. Facial expression recognition is usually performed in four steps, including pre-processing, face detection, feature extraction, and expression classification.

In this project, we have used a variety of intensive deeep learning techniques (convolutional neural networks) to identify the main seven human emotions: anger, hate, fear, joy, sadness, surprise, and neutrality.

I. Neutral II. Angry III. Disgust IV. Fear V. Happy VI. Sadness VII. Surprise



1. INTRODUCTION

Human facial expressions are categorized into 7 universal emotions: happy, sad, surprised, terrified, angry, ugly, and neutral. Our facial emotions are expressed through the activation of specific sets of facial muscles. These are sometimes subtle and complex signs of expression that contain abundant

information about the state of our mind. Through this, we can measure impacts and measurements on audiences / customers with ease and cost.

For example

- Retailers can use these metrics to assess customer interest.
- Healthcare providers can provide better services by using additional information about patients' mood during treatment.
- Entertainment producers can continually monitor audience engagement at events to create the desired content.

The Humans are well-trained to read the feelings of others, in fact, at just 14 months of age, children can already tell the difference between happy and sad. Computers work better than us in getting to the mood of humans. Therefore, we have designed/created an intensive deep learning neural network that gives machines the ability to communicate about our emotional states. In other words, we give eyes to the system for what we can see.

2. PROJECT FORMULATION

The hands on building this project of Facial Expression Recognition is divided into following tasks/steps:-

A. Task 1: Introduction

- Introduction to the dataset
- Import essential modules and helper functions from NumPy, Matplotlib, and Keras.

B. Task 2: Exploring the Dataset

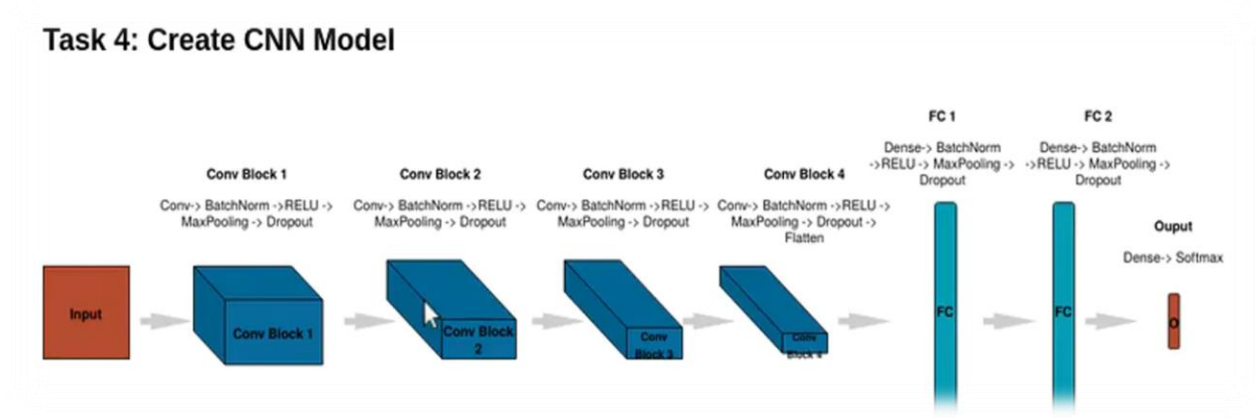
- Display some images from every expression type in the Emotion FER dataset.

C. Task 3: Generating Training and Validation Batches

- Generate batches of tensor image data with real-time data augmentation.
- Specify paths to training and validation image directories and generates batches of augmented data.

D. Task 4: Creating a Convolutional Neural Network (CNN) Model

- Design a convolutional neural network with 4 convolution layers and 2 fully connected layers to predict 7 types of facial expressions.
- Used Adam as the optimizer, categorical crossentropy as the loss function, and accuracy as the evaluation metric.



E. Task 5: Training and Evaluating Model

- Training the CNN by invoking the **model.fit()** method.
- Used **ModelCheckpoint()** to save the weights associated with the higher validation accuracy.
- Observed live training loss and accuracy plots in Jupyter Notebook for Keras.

F. Task 6: Saving and Serializing Model as JSON String

- Used **to_json()**, which uses a JSON string, to store the model architecture.

G. Task 7: Creating a Flask App to Serve Predictions

- We used the open-source code from "Video Streaming with Flask Example" to create a flask app to serve the model's prediction images directly to a web interface.

H. Task 8: Creating a Class to Output Model Predictions

- Created a FacialExpressionModel class to load the model from the JSON file, load the trained weights into the model, and predict facial expressions.

I. Task 9: Designed an HTML Template for the Flask App

- Designed a basic template in HTML to create the layout for the Flask app.

J. Task 10: Used Model to Recognize Facial Expressions at the Real Time using laptops webcamera

- We than run the **main.py** script to create the Flask app and serve the model's predictions to a web interface.
- Applied the model for real time recognition of facial expresssions of users using webcam of the Laptop.

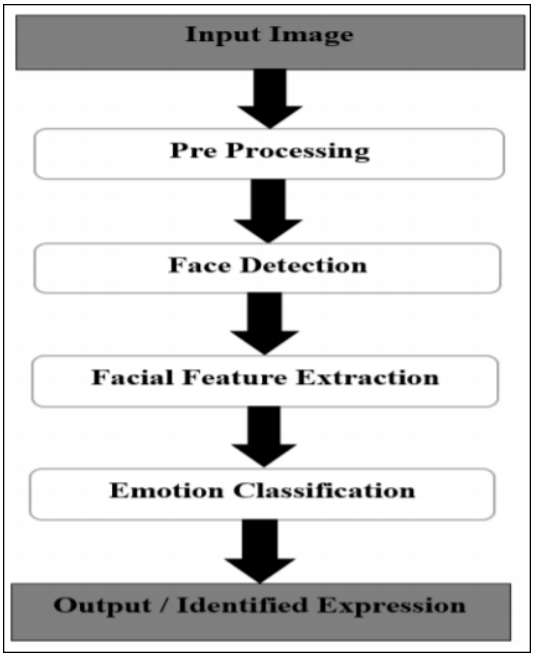
3. DATASET DESCRIPTION

The dataset used in this project work has been taken from the Kaggle.com available at (https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data). The data consists of 48x48 pixel grayscale images of faces. The faces have been automatically registered so that the face is more or less centered and occupies about the same amount of space in each image. The task is to categorize each face based on the emotion shown in the facial expression in to one of seven categories (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral).

train.csv contains two columns, "emotion" and "pixels". The "emotion" column contains a numeric code ranging from 0 to 6, inclusive, for the emotion that is present in the image. The "pixels" column contains a string surrounded in quotes for each image. The contents of this string a space-separated pixel values in row major order. test.csv contains only the "pixels" column and your task is to predict the emotion column.

The training set consists of 28,709 examples. The public test set used for the leaderboard consists of 3,589 examples. The final test set, which was used to determine the winner of the competition, consists of another 3,589 examples.

4. PROJECT STRUCTURE



Facial expression recognition is a process performed by humans or computers, which consist of:-

1. **Locating faces in the scene** (e.g., in an image; this step is also referred to as facedetection),

2. **Extracting facial features** from the detected face region (e.g., detecting the shape of facialcomponents or describing the texture of the skin in a facial area; this step is referred to asfacial feature extraction),

3. **Analyzing the motion of facial features** and/or the changes in the appearance of facialfeatures and classifying this information into some facial-expressioninterpretativecategories such as facial muscle activations like smile or frown, emotion (affect)categories like happiness or anger, attitude categories like (dis)liking or ambivalence, etc.(this step is also referred to as facial expression interpretation).

5. STEPS FOLLOWED FOR BUILDING THE PROJECT

As per various surveys it is found that for implementing this project four basic steps are required to be performed.

i.) Preprocessing

ii.) Face registration

iii.) Facial feature extraction

iv.) Emotion classification

Description about all these processes are given below-

1. Preprocessing : Preprocessing is a common name for operations with images at the lowest level of abstraction both input and output are intensity images. Most preprocessing steps that are implemented are – a. Reduce the noise b. Convert The Image To Binary/Grayscale. c. Pixel Brightness Transformation. d. Geometric Transformation.

2. **Face Registration** : Face Registration is a computer technology being used in a variety of applications that identifies human faces in digital images. In this face registration step, faces are first located in the image using some set of landmark points called “face localization” or “face detection”. These detected faces are then geometrically normalized to match some template image in a process called “faceregistration”.

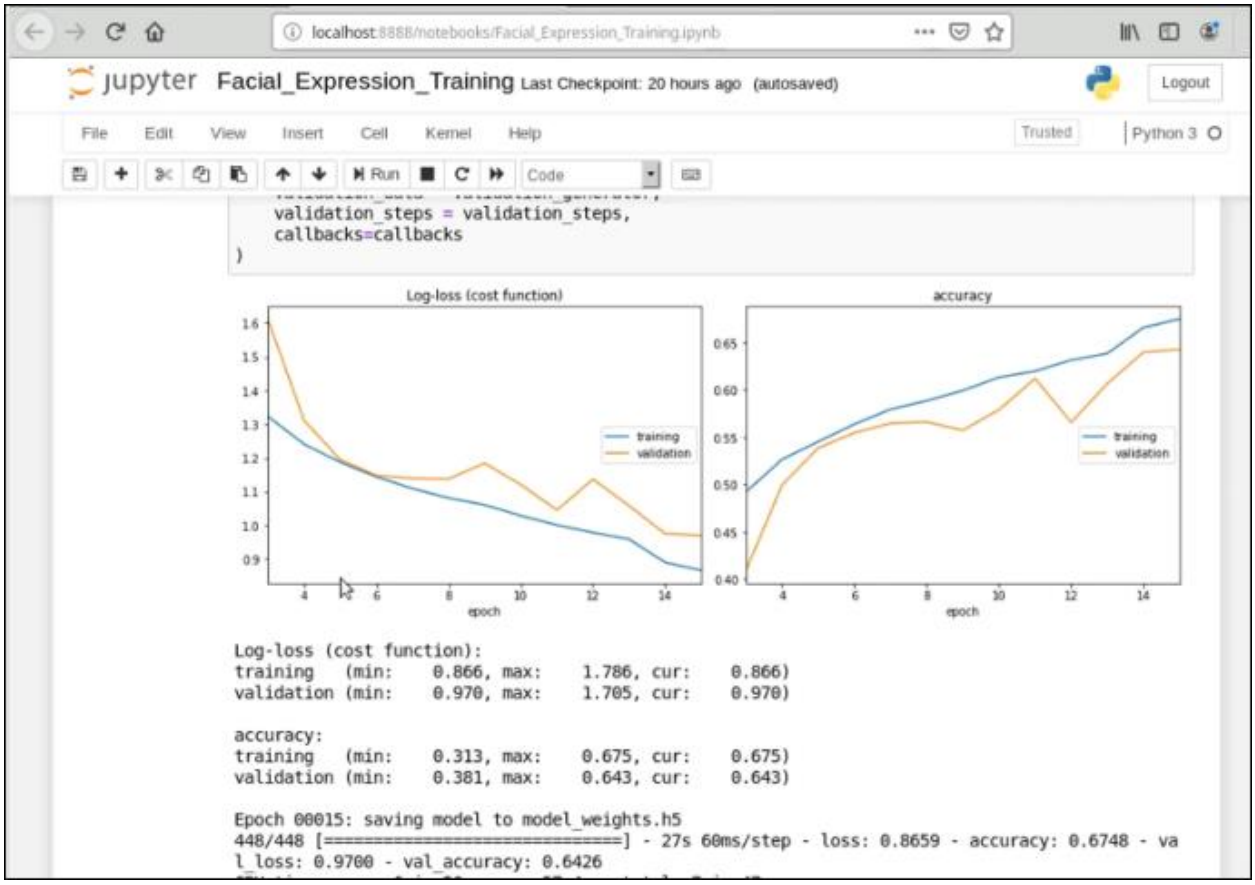
3. Facial Feature Extraction : Facial Features extraction is an important step in face recognition and is defined as the process of locating specific regions, points, landmarks, or curves/contours in a given 2-D image or a 3D range image. In this feature extraction step, a numerical feature vector is generated from the resulting registered image.

Common features that can be extracted are:- a. Lips b. Eyes c. Eyebrows d. Nose tip

4. Emotion Classification : In this step, of classification, the algorithm attempts to classify the given faces portraying one of the seven basic emotions.

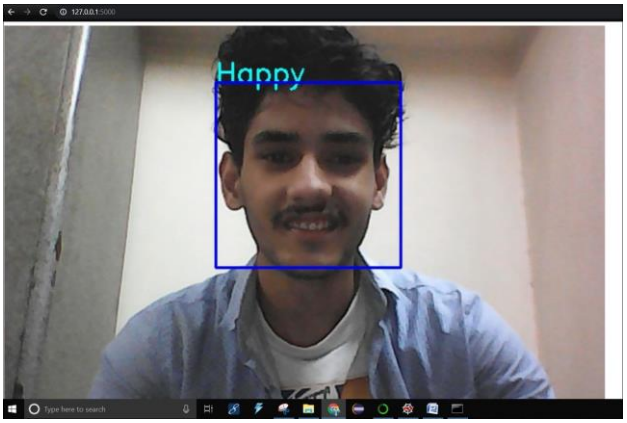
6. IMPLEMENTATION

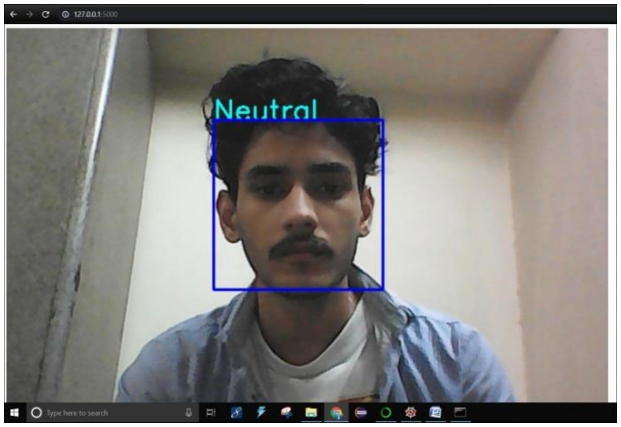
- **Accuracy and Loss of the Model:-**

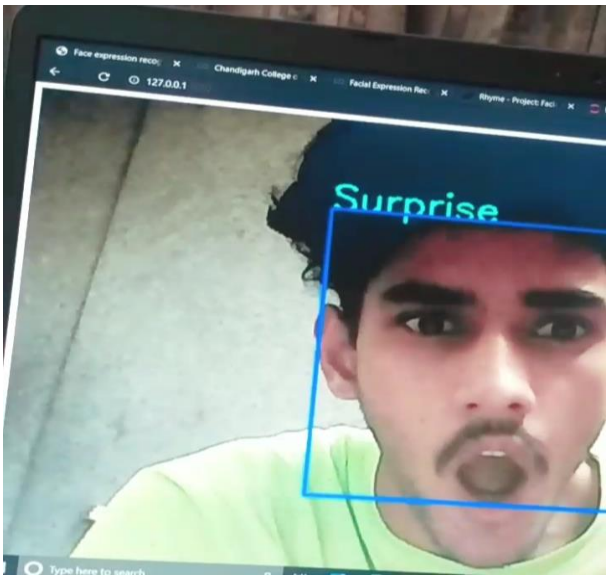


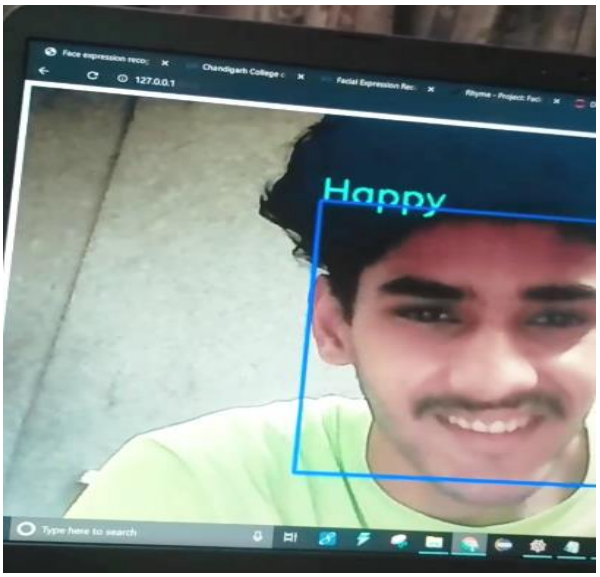
7. OUTPUT SCREENSHOTS

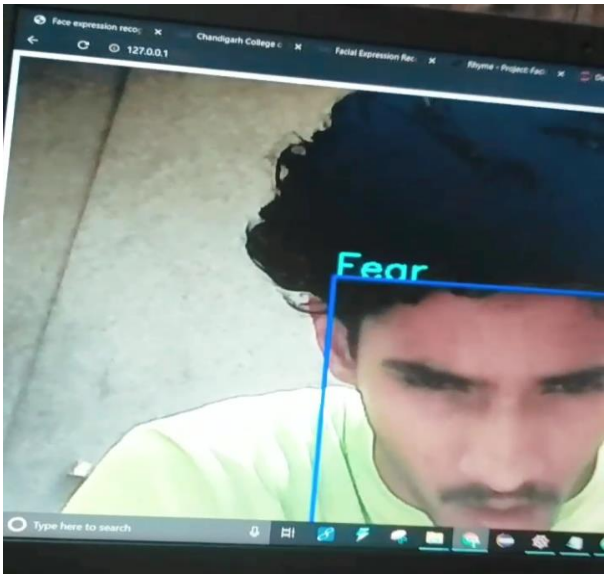
- **Prediction of Facial Expressions on Web Interface**

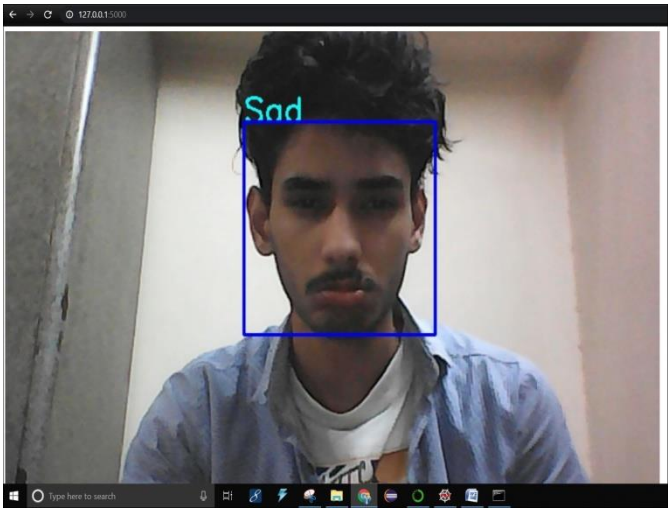












8. SOFTWARE REQUIREMENTS

As the project is developed in python, we have used Anaconda for Python 3.6.5 and Spyder.

Anaconda

Anaconda is a free and open source distribution of the Python and R programming languages for data science and machine learning related applications (large-scale data processing, predictive analytics, scientific computing), that aims to simplify package management and deployment. Package versions are managed by the package management system conda. The Anaconda distribution is used by over 6 million users, and it includes more than 250 popular data science packages suitable for Windows, Linux, and MacOS.

Jupyter

The Jupyter Notebook is an open source web application that we can use to create and share documents that contain live code, equations, visualizations, and text. Jupyter Notebooks are a spin-off project from the IPython project, which used to have an IPython Notebook project itself. The name, Jupyter, comes from the core supported programming languages that it supports: Julia, Python, and R. Jupyter ships with the IPython kernel, which allows us to write our programs in Python, but there are currently over 100 other kernels that we can also use.

Spyder

Spyder (formerly Pydee) is an open source cross-platform integrated development environment (IDE) for scientific programming in the Python language. Spyder integrates NumPy, SciPy, Matplotlib and IPython, as well as other open source software. It is released under the MIT license. Spyder is extensible with plugins, includes support for interactive tools for data inspection and embeds Python-specific code quality assurance and introspection instruments, such as Pyflakes, Pylint and Rope. It is available cross-platform through Anaconda, on Windows with WinPython and Python (x,y), on macOS through MacPorts, and on major Linux distributions such as Arch Linux, Debian, Fedora, Gentoo Linux, openSUSE and Ubuntu.

Atom (Text and Source Code Editor)

Atom is a free and open-source text and source code editor for macOS, Linux, and Microsoft Windows with support for plug-ins written in Node.js, and embedded Git Control, developed by GitHub. Atom is a desktop application built using web technologies. Most of the extending packages have free software licenses and are community-built and maintained. Atom is based on Electron (formerly known as Atom Shell), a framework that enables cross-platform desktop applications using Chromium and Node.js. It is written in CoffeeScript and

Less.Atom was released from beta, as version 1.0, on 25 June 2015. Its developers call it a "hackable text editor for the 21st Century". It is fully customizable in HTML, CSS, and JavaScript.

We used this software to build our HTML page for the web interface of our project.

9. CONCLUSION

In this project a Emotion/Facial Recognition model has been trained and saved. It can recognize/detect the facial expressions of an individual on a real time basis that whether the individual is Neutral, Angry, Disgust, Fear, Happy, Sad, Surprised.

REFERENCES

[1] https://www.coursera.org/learn/facial-expression-recognition-keras/supplement/2KrW0/ project-based-course-overview

[2] https://www.kaggle.com/ashishpatel26/tutorial-facial-expression-classification-keras

[3] Fundamentals of Facial/Emotion Recognition